Thesis for The Degree of Licentiate of Engineering

Requirements Engineering Challenges of Supporting Agile Teams in System Development

Rashidah Kasauli Namisanvu

Division of Software Engineering
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2018

Requirements Engineering Challenges of Supporting Agile Teams in System Development

Rashidah Kasauli Namisanvu

Technical Report No 190L
ISSN 1652-876X
Department of Computer Science & Engineering
Division of Software Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2018.

To my family

Abstract

Context: Agile methods have attracted many companies due to their reported benefits of short time-to-market and improved quality outputs. In the systems development context, additional constraints apply e.g. as a result of scale or parallel development of hardware and software. Traditionally, stage-gate processes with a focus on up-front requirements analysis are common in large-scale systems engineering. These processes clash with the companies' desire to become more agile.

Objective: The aim of this thesis is to discover challenges that new requirements engineering approaches should address to enable agile system development at scale (RE4Agile). With a focus on value and building system understanding, we explore these challenges from the perspective of the agile development teams.

Method: To meet our aim, we conducted a series of empirical studies based on case studies, and a secondary review to explore the problem domain while deriving challenges and potential solutions from industry and literature respectively.

Findings: Our findings show that there are numerous challenges of conducting requirements engineering in agile development especially where systems development is concerned. These challenges relate to user value and overall system understanding. However, there are some cross-cutting concerns, e.g safety-critical development, that have generated much interest both from practitioners and academicians at large.

Conclusions: The challenges discovered sprout from an integration problem of working with agile methods while using the already existing processes as well. However, solution candidates exist and our future research aims to validate some of the solution candidates in the view of deriving new RE approaches. This thesis contributes to such future research, by establishing a holistic map of challenges that allows to assess whether a given solution is beneficial in the larger context or whether it over-optimizes only one area.

Keywords

Scaled-Agile System Development, Requirements Engineering, Safety-critical System Development, User Value

Acknowledgment

My heartfelt gratitude goes to my main supervisor Eric Knauss for first of all accepting to be my supervisor. He has gracefully taught me in this new life from the social to the academic. I am what I am now because he is that good!. Thank you for the good advice and friendly environment you always provide. You overlooked my playful nature and taught me to look at life in a much more beautiful fashion. Thank you!

To my co-supervisor supervisor, Benjamin Kanagwa thank you for the wonderful feedback whenever I needed it. I know I can be stressing and sometimes nagging, but you always find ways to calm me down. I appreciate that.

I also thank colleagues in the Software Center Project 27 for the continued support and advice. Special thanks goes to Grischa Liebel, Jennifer Horkoff and Francisco Gomes. I want to extend my gratitude to M.R. V. Chauldron and Engineer Bainomugisha for always making sure we never loose sight of the overall project objective.

I thank my family, husband and children for persevering for this time so far and giving me the encouragement I need to meet my aims in the required time. May God bless you all.

List of Publications

Appended publications

This thesis is based on the following publications:

[A] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar and B. Kanagwa "Requirements Engineering Challenges in Large-Scale Agile System Development"
In IEEE 25th International Requirements Engineering Conference (RE) pp. 352-361, 2017.

[B] R. Kasauli, J. Nakatumba-Nabende, B. Kanagwa "Agile Islands in a Waterfall Environment: Challenges and Strategies"
In submission to REFSQ, 2019

[C] R. Kasauli, E. Knauss, A. Nilsson, S. Klug "Adding Value Every Sprint: A Case Study on Large-Scale Continuous Requirements Engineering"
In 3rd Workshop on Continuous Requirements Engineering, REFSQ Workshops, 2017.

[D] R. Kasauli, E. Knauss, B. Kanagwa, A. Nilsson, G. Calikli "Safety-Critical Systems and Agile Development: A Mapping Study"
In 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp.470-477. IEEE 2018.

Other publications

The following publications were published during my PhD studies. However, they are not appended to this thesis, due to contents not related to the thesis.

[a] R. Kasauli. "Requirements Engineering for Large Scale Agile Systems Development."
In REFSQ Workshops (Doctoral Symposium), 2017.

[b] E. Knauss, G. Liebel, K. Schneider, J. Horkoff, and R. Kasauli. "Quality Requirements in Agile as a Knowledge Management Problem: More than Just-in-Time."
In IEEE 25th International Requirements Engineering Conference Workshops (REW), pp. 427-430, 2017.

[c] F.G. de Oliveira Neto, J. Horkoff, E. Knauss, R. Kasauli, and G. Liebel "Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study."
In IEEE 25th International Requirements Engineering Conference Workshops (REW) pp. 315-322, 2017.

[d] E. Knauss, G. Liebel, J. Horkoff, R. Wohlrab, R. Kasauli, F. Lange, and P. Gildert. "T-Reqs: Tool Support for Managing Requirements in Large-Scale Agile System Development."
In IEEE 26th International Requirements Engineering Conference Workshops (REW), 2018.

[e] R.Kasauli et al. "Requirements Engineering Challenges and Practices in Large-Scale Agile Systems Development"
To be submitted to REJ

Research Contribution

The included papers were published with collaborations from colleagues. As main author to all the included papers, I was responsible for the research design. In particular, I have contributed in the included papers as follows.

In Paper A, I participated in all phases of data collection and analysis. In addition, I coordinated and participated in the paper writing. In paper B, the data was collected by my co-authors. I was responsible for the paper design and most of the writing. In papers C and D, the design, planning, execution of the research and writing of the papers was mostly done by me.

Contents

1 Introduction

Software has pervaded our lives and is continuously gaining importance. Nowadays, even the formally hardware based systems like cars are becoming more software oriented than before [1]. This increased use of software has led to more software-intensive systems, i.e. systems that consists of software, hardware and possibly mechatronic parts defining the context in which they are used. Examples of such large-scale systems include communications, entertainment and automotive systems For such software-intense systems, requirements engineering is the key to success [2, 3].

Requirements Engineering (RE), the process of gathering and defining user expectations for a new or modified product, is traditionally a sequential process where execution of, for instance software development requires indisputable completion of the requirements specification phase [2, 4]. This approach to RE is the foundation on which most traditional companies are built, including most large-scale systems development companies. With advancement in software, and new players coming into the market, competition has increased and customer demands are evolving much faster, making reliance on traditional methods, with their long lead times, less focus on customer value, and lack of flexibility, less of an option. Thus large-scale systems development companies are seeking better approaches that allow flexibility, a characteristic of agile development methods.

Agile development methods become increasingly attractive for developing software-intense large systems as they have significantly contributed to the way software is developed [5]. Driven by reported success in handling changing customer demands, achieving shorter time-to-market and improved quality outputs [6], large-scale companies are also adopting the agile methods [7–9]. However, the agile methods were originally meant for small teams [10] making their adoption at scale challenging, not only because of the scale but also that many large-scale companies are founded on traditional methods and thus could not just do a sudden overhaul but rather a sequential adoption [11–13].

Agile development promotes customer collaboration, thus, RE and Agile seem to support each other. However, long upfront analysis is considered anti agile and there is some friction between RE (which is often considered as anti-agile or traditional) and agile methods. Also, although it is clear how to perform RE in a traditional context, it is not clear to how RE can be done in an agile environment.

This thesis presents the initial results towards attempts to address this tension. We approach the problem through a series of empirical studies (Paper A, B and C) that discover the information needs and related knowledge,

pertinent to product development and finally focus on safety critical systems development, through a mapping study (Paper D).

We started with an exploratory study (Paper A), to uncover the RE related challenges of agile development in large-scale companies and had the following contributions:

- We discovered that three out of the four companies we worked with had agile development only on team level while the rest of the organization was still plan-driven. The revelation instigated a study that led to investigation of what challenges the agile teams face while interfacing with traditional structures (Paper B) to which we found challenges relating to information/knowledge transfer for better system understanding.

- We found challenges relating to user value and system understanding while using agile development at scale. Since agile methods are about delivering of value to the customers, to give us better ground to investigate and propose solutions for the identified challenges, we also conducted a study to find out the teams' interpretation of customer value (Paper C). We found that there was lack of shared understanding of customer value which could impede continuous delivery and deployment. We thus highlight the need for continuous requirements engineering practices.

- The two areas of requirements knowledge obtained in Paper A, i.e user value and system understanding, are in line with traditional practices but not particularly supported in agile development. Paper A identified different examples that explain that phenomenon including impact on infrastructure, safety-critical and agile. However, preliminary search on safety-critical and agile development presented the least empirical studies out of all the obtained categories, yet it is a concern for software-intensive large-scale organizations. So we conducted a mapping study (Paper D) on the use of agile methods in safety-critical systems development. We focus on the benefits, challenges, principles and practices used. We conclude with a need for guidelines on how to shift effort to just-in-time activities and also how to establish beneficial infrastructure and long-term support. With this we would address the development process of safety systems in an agile environment.

Overall, this thesis set out to understand the challenges of performing agile development in large-scale systems development contexts and suggest candidate solutions with the intention of providing more empirical insights into the field of RE.

The thesis is composed of two parts, the introduction part (Chapter 1) and the second part is an attachment of the included papers. The rest of this introduction chapter is structured as follows: Section 1.1 presents the background and related work of the research presented in this thesis. Section 1.2 presents our research questions. The research methodology is described in Section 1.3 while Section 1.4 provides a synthesis of our research outputs. In Section 1.5, we give the conclusions and future work. For the second part, we have Chapter 2 to Chapter 5 with Papers A - D respectively.

1.1 Background and Related Work

The adoption of agile methods has changed the way RE is interpreted in development. Whereas some argue that RE can be viewed from two different angles; 1) as a formal and structured transformation of information [14] (e.g. traditionalists) or 2) as a collaborative effort relying on the creativity and competence of the involved engineers [15] (e.g. the agilists), others seem to imply that RE seized to exist with the introduction of agile methods (e.g. 'architecturalists'). In this thesis, we take the view that RE is a collaborative effort to which agile methods is an example of. In this section we discuss the RE process in systems development in which we describe the traditional RE process and fundamental RE terminology. We then review the literature on agile software development with a focus on RE in agile in order to provide basic understanding of that domain. We then discuss concepts related to user value, system understanding and finally cross-cutting concerns in agile systems development.

1.1.1 RE in Systems Development

Systems development or engineering was initially about configuring hardware components into physical systems like ships or railroads [16]. The component parts would then be produced once the configuration and the requirements specification are done. Software then began to appear in such systems. Systems' dependence on software has increased over the years with more software being used in e.g. automotive [1, 17]. Thus competitive advantage is increasingly depending on software as it facilitates rapid tailoring of products and services to market demands [16]. The dependence on software has led to software-intensive systems - systems that depend on software, hardware and the context in which they are operating for correct operations. And in such systems, effective requirements engineering is the key to success [4].

With the recent reports of software failures being associated with requirements challenges [3], organisations have to perform effective RE in order to keep pace with complexities [18] and beat time to market with a quality product that meets customers' needs. However, when software components began to appear in systems development, sequential development then came naturally [16, 19] making RE to be traditionally seen as a set of sequential activities. The activities involved include requirements elicitation, analysis, specification, and validation in that order, with requirements prioritisation coming in to support elicitation and analysis by identifying the most valuable requirements [20].

Requirements elicitation process includes users getting involved in gathering requirements [21]. During elicitation, the initial information regarding requirements and context are gathered. The *requirements analysis* phase then follows to check for consistency, completeness, necessity and feasibility of the requirements, thus creating an understanding of the requirements. The next activity is the *requirements specification* where the requirements are defined in terms of system behaviour, decomposing the problem into component parts and serve as input to design specification. The end of this process is marked with a requirements specification document where the agreed requirements are documented for communication with stakeholders and developers. The *require-*

ments validation is important for confirming customers' needs and correcting errors in the specifications to avoid rework which could be expensive.

With increased emphasis on users and end value as well as increased pace of change, more strain has been put on the traditional approach. The strain came from the realization that requirements were more emergent with use than pre-specifiable, and the traditional methods were not suitable for producing user valued products [16]. Alternative RE methods, like agile methods had to be devised and adopted.

1.1.2 Agile RE or is it RE in Agile Development?

The agile alliance [22] gives the following definitions for agile and agile software development;

- Agile as "the ability to create and respond to change in order to succeed in an uncertain and turbulent environment".
- Agile software development as an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto [23].

The agile manifesto identifies four values for agile development as follows:

Table 1.1: The four values for agile development

1.	Individuals and interactions	over	processes and tools.
2.	Working software	over	comprehensive documentation.
3.	Customer collaboration	over	contract negotiation.
4.	Responding to change	over	following a plan.

While the agile advocates acknowledged the items on the right as having value, they valued the items on the left more (see Table 1.1). The agile manifesto also connect 12 principles that attempt to make the agile values more concrete and deliver solid guidance for software development teams and their projects. The agile principles are originally as presented in Table 1.2. These have been reviewed by William [24] but the basic concepts remain the same.

Agile methods like Scrum [25] and XP [26] are thus based on the above values and principles and encourage flexible and light-weight software development with short iterations [5] thus creating ability to deal with changing requirements and fast time-to-market. In agile software development, requirements are allowed to evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.

Agile RE or RE for agile development has no unanimously accepted definition [19] but can be weakly defined as the agile way of performing RE. Bjarnasson *et al.* [27] report that agile methods address some classical RE challenges like communication gaps but also cause new challenges. Existing research on agile RE has explored the practices used [15, 21] and agreed on some practices that apply to agile RE, including; *face-to-face communication, customer involvement, requirements prioritization, review meetings and retrospectives, iterative/incremental development, user-stories, test driven development, acceptance tests, change management and code refactoring.* The practices adopted

Table 1.2: The 12 Agile Principles [22]

No.	Principle
1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4.	Business people and developers must work together daily throughout the project.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7.	Working software is the primary measure of progress.
8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9.	Continuous attention to technical excellence and good design enhances agility.
10.	Simplicity – the art of maximizing the amount of work not done – is essential.
11.	The best architectures, requirements, and designs emerge from self-organizing teams.
12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

by teams varies depending on the agile method of development that is chosen. For instance, in XP, the planning game is used and begins with an on-site customer who writes the requirements. These are later prioritized by the development team together with the customer and so on. However, experience with practitioners has shown that practices and methods have been adopted with rather random choice according to the development needs. Some existing works have identified challenges of using agile RE [15,21] that include neglect of non-functional requirements, customer availability and minimal documentation. Also, although agile development is about value delivery, it is not entirely clear how it is interpreted by the development teams.

In summary, there is substantial amount of existing work on the use of agile RE and its practices but a lack of empirical evidence in terms of systems development in a large-scale software-intensive context. It is also not clear how agile RE addresses the concepts of user value and building system understanding.

1.1.3 User Value

Today, software has a major influence on systems' costs and value, and software decisions are intertwined with system-level decisions [28]. Being more of a value-

driven development approach [23], agile methods offer a solution to handling value during development and there exists research that discusses the creation of customer value as key element for a company's success [29–31].

Related to this concept of customer value, Boehm [28] proposes a focus on value-based software engineering (VBSE) approaches that include, among others, value-based requirements engineering (VBRE). According to Boehm, VBRE includes principles and practices for identifying systems' success-critical stakeholders, eliciting their value propositions with respect to the system and reconciling these into mutually satisfactory objectives of the system [28]. Thus VBRE uses a selection of requirements to enhance the value of a release [31, 32]. Through VBRE, development teams are able to align customers requirements, business requirements, and technological opportunities, to have a sound understanding of both technical and business implications of decisions made, and to understand the business dynamics that drive software development [33].

However, even with such defined approaches, the different interpretations of customer value can inhibit value inclusion in development [34]. Customer value is frequently related to the trade-off between what the customer receives and what they invest to acquire and use a product [35]. This definition is based on the customer's perspective, and to our knowledge there exists little research on how software development teams can relate to customer value, especially in large-scale systems development. Alahyari *et al.* investigate how value is interpreted, prioritized, assured, and measured in agile software development [36]. Based on a qualitative study with 23 participants from 14 organizations, they identified and prioritized value aspects such as *delivery process with respect to time*, *quality*, and *knowledge of feature value for customer*. In a position paper, Kuusinen discusses the meaning of value for business owners, customers, users, software developers, and user experience specialists and works towards an understanding on how to align and articulate value and its delivery in a software project [34].

1.1.4 System Understanding

Software engineering is a knowledge-intensive endeavor [37, 38] with activities from requirements elicitation to the project coordination and management. It is unlikely for the team members to have all the knowledge obtained from those activities [38]. Agile development adds to the challenge with the idea of breaking down the system requirements into features to be developed every sprint. Thus making the bigger picture of system understanding unclear. However, in order to deliver a quality product, the development team has to have a clear understanding of the system. Effective communication and knowledge management become an essential part of their development.

Knowledge management is crucial for proper system understanding and ensuring effective communication helps to transfer knowledge [39]. However, there are few existing works that explicitly address communication in agile development [40]. Communication in agile projects has mainly focused on the impact of the agile practices on communication [41]. In a systematic review, Hummel *et al.* found reports that agile methods lead to improved communications in large-scale development projects. They also found that the

informal communication on which agile methods depend may be problematic for large projects with many stakeholders and a lot of shared information. Studies suggest synchronous and asynchronous communication means e.g. wikis and group e-mails in order to establish the multiplicity of social links between team members and to provide continual access to project information in large-scale settings [40]. The studies present a broad understanding of the communication concept without focusing on the social interaction and behavior of teams [40].

1.1.5 Cross-cutting concerns in Development

We use the term cross-cutting concerns to mean the development concerns that seem to be affected by both user value and system understanding. These are usually the quality concerns or the non-functional requirements of any software development. From the research conducted in agile software development, there are challenges of dealing with non-functional requirements [42]. However, since agile methods tend to have less favor for documentation and processes, the most affected cross-cutting concern then becomes safety [43, 44] since this requires a well defined process with extra documentation for certification.

1.2 Research Questions

The goal of this thesis is to provide an empirical exploration into the challenges of using agile methods in the development of software-intensive systems from a requirements engineering perspective. In order to achieve our goal, we focus on large-scale systems development companies, considering that the use of agile methods suffers most when used at scale. Also the large-scale systems have many of safety-critical features and safety is known to be driven by requirements for quality development and thus we take on a requirements engineering perspective. We therefore defined the following main research question:

RQ: What challenges are associated with agile development of software-intensive systems from a requirements engineering perspective? Understanding what to put in place in such environments so teams can be effective.

The answer to this overarching research question is intended to provide insight into the RE challenges of using agile development at scale, so that we can understand what to put in place to ensure team effectiveness and efficiency. In order to answer the main research question while scoping the thesis, three refined research questions were derived and addressed in the included four studies/papers. These refined research questions are presented in Table 1.3

The four research questions are designed to gain insights into the challenges and possible practices in the problem domain. We start with RQ1 which is more of a general question for the wider understanding of the problems in RE for agile development. While this gave a wide scope of challenges, we explored the ones that geared interest for the participating companies. Thus RQ2 set out to further scope our problem by identifying the challenges of having teams doing agile development in an environment that is structured. For RQ3, since agile development is about creating value for the customer, we set out to understand how the agile teams interpret value during development. To top up the exploration, RQ4 sets out to explore the challenges that teams face while

Table 1.3: Research Question and Paper in which it is addressed.

	Research Question (RQ)	Paper
RQ1	What are the RE challenges of using agile methods in large-scale systems development?	Paper A
RQ2	What challenges do agile teams face while developing in a plan-driven environment?	Paper B
RQ3	How is value interpreted/viewed by the development teams in agile development?	Paper C
RQ4	What are the challenges of managing cross-cutting concerns in an agile environment (e.g. developing safety-critical systems)?	Paper D

developing safety-critical systems using agile methods. The research questions are interconnected and answered in different papers as shown in Figure 1.1.

1.3 Research Methodology

This research aimed to investigate how the software engineers (development teams) conduct their development and other operations while using agile methods in a large-scale context. Being that software development is carried out by persons or groups in organizations [45], it is a multi-disciplinary area that also includes the social boundaries. Thus for this research, we not only need to investigate the tools and processes used by the development teams, but also their social and cognitive processes [46]. Thus the use of empirical methods.

Being more geared towards exploring the problem domain, this thesis mainly consists of empirical studies (Papers A-C) using the case study methodology and one review study (Paper D). This section elaborates on the methods used and goes ahead to elaborate how it was used in the included studies.

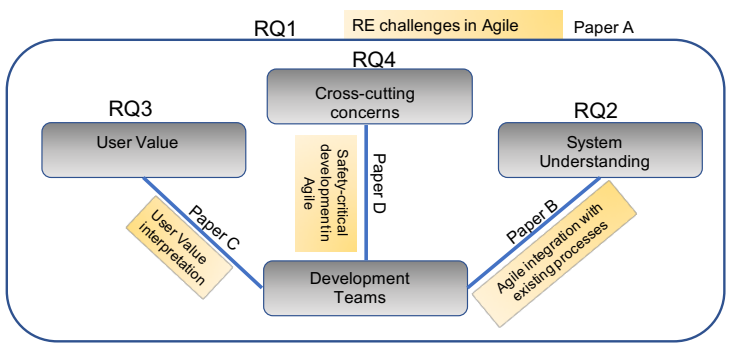

Figure 1.1: Overview of studies in relation to research questions.

1.3.1 Case Study Method

According to Runeson *et al.*, case study is *an empirical inquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified* [47], which is also the aim of this thesis. Case studies provide in-depth understanding of why and how given phenomenon occur [46]. This thesis is based solely on exploratory studies which are commonly used as initial investigations to seek new insights [47] and possibly identify useful distinctions that clarify our understanding [46].

Runeson and Host [45] present five steps to performing case study research and these include: (i) designing case study objectives and plan (ii) Defining data collection procedures and protocols (iii) Collecting data on the studied case, (iv) analysing the data and (v) reporting the data. For the first step, we had our research questions prepared before starting the study and purposely [48] identified the cases to use in our studies. For all the studies, we chose large-scale software intensive companies that have agile development teams. While identifying the companies, we also discussed with the contacts in the different companies on the acceptable data collection procedures that will bring out the best for both parties.

Case studies can be holistic where the case study is the unit of analysis, or embedded where one can either have one case study with many units of analysis or many (different) case studies. We used all those set-ups in the different studies since the studies were targeting different questions. For instance, in Paper A we aimed to find out the overall challenges of doing RE in an agile development company and thus we chose to use a multiple case study approach where we had four cases understudy. The cases were all from different domains. This choice helped us discover as many challenges as there were and we could attempt to generalize when we find challenges reoccurring in different domains. The findings of Paper A inspired the study that led to Paper B, C and D. For paper B, we used a single case study with two units of analysis as we aimed to find the challenges that individual agile teams faced working in a structured environment. The single case was viable since it provides more focus and the two units of analysis were used for comparison of findings. Paper C was single case study since we aimed to find out how the agile team interpreted user value during their development. It provided a basis for solution derivation.

1.3.1.1 Data collection

The data for the three case studies included was collected through use of semi-structured interviews, workshops and focus group meetings. Depending on the research question being posed, the choice of data collection methods changed considerably. For instance in Paper A, we used the combination of all three techniques where we had 2 cross-company workshops attended by representatives from all the case companies, 22 interviews with the development teams from three of the companies, and 5 focus group meetings with 4 of the participating companies. Figure 1.2 presents a summary of the included papers and the data collection methods used. During these data collection activities, we would have at least two of the researchers present to take notes and compare

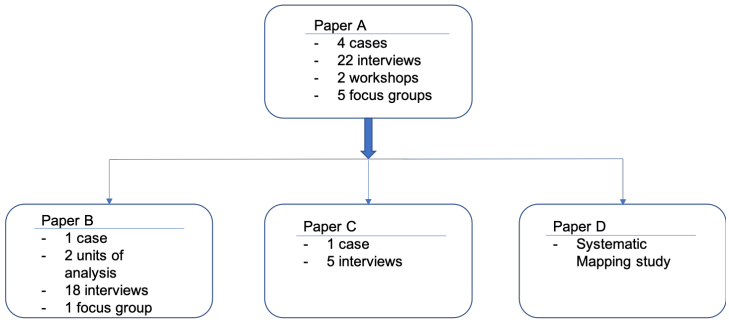

Figure 1.2: Case studies and data collection methods used.

afterwards. This helped us to mitigate possible bias and misunderstanding.

1.3.1.2 Data analysis

The data collected in this thesis is qualitative because qualitative data is appropriate to fulfill our research goal. For that reason, we collected the data using interviews, focus group meetings and workshops during which responses were recorded and later transcribed thus leading to textual analysis. For the data analysis, we relied on thematic coding approach [49]. Since we worked in groups, we had at least two researchers at each study to familiarize themselves with the data collected while highlighting noteworthy statements and assigning codes or labels to each. We then discuss as a group and iteratively agree on the themes which we discuss through workshops (Paper A) or through email and telephone calls (Paper B and C) for validation with the participating cases.

1.3.2 Secondary Study

With the growing number of empirical studies in software engineering comes a necessity to construct an objective summary of the available research evidence to aid in decision making and formulation of research questions [50]. Using a systematic literature review is one way of obtaining the objective summary. Much as each study included in this thesis has a literature review section, a systematic review provides guidelines to follow while reviewing literature in order to avoid bias and ensure replicability [51]. Systematic literature reviews however require considerable amount of effort [52].

A systematic mapping study provides a map of the results reported in literature, usually a more coarse overview thus often requires less effort than a systematic literature review [51]. The process followed in a mapping study is also systematic but more coarse than a systematic literature review, allowing to process larger numbers of papers.

Following the findings from Paper A and an unclear course of action for answering RQ4, the study of safety-critical development in agile, we chose to conduct a mapping study (Paper D) to help us draw a map on the current state of affairs in as far as safety-critical systems development is concerned. We aimed to find out what the benefits, challenges, principles and practices being used in industry and thus we relied more on the empirical studies in that

field. The relevant literature was searched from the Elsevier Scopus database and categorized according to the research questions.

1.3.3 Threats to validity

For this thesis, as with many empirical studies, there are validity threats worth discussing. We consider the four perspectives of validity threats as presented in Runeson and Host [45]and in Easterbrook *et al.* [46].

1.3.3.1 Construct validity

These threats refer to the relations between the research method and the observations from the study [45]. With these threats the question to answer is: Are the theoretical constructs interpreted and measured correctly? There is a threat that the interpretation of the questions asked at the interviews may be different for the researchers and the interviewees due to the use of different terms. In order to minimise this threat, we shared and discussed the interview guide with the company contacts in order to agree to commonly understood/used terms at the company and also used literature to provide a link between our understanding and that of the interviewees. In cases where the interviewee did not understand the question, we endeavored to rephrase. We also ensured that we had more than one researcher for data collection and analysis in all the studies. To combat the practitioners' fear to answer asked questions with honesty, we guaranteed anonymity and answers were only to be used by the researchers.

1.3.3.2 Internal validity

Focuses on the research design and whether the results really do follow from the data [46]. Could external factors impact the results of the investigated factors? To minimise this risk, we recorded our interviews so that each researcher gets the same message. To increase internal validity for our studies, we used data triangulation between interviews (Paper A,B, and C), between the units of analysis (Paper B), and between the case companies (Paper A). Furthermore, the results of our studies were discussed at workshops (Paper A and C) and at focus group meetings (Paper B). The workshops and focus groups included key roles from these companies that were already involved in the respective studies. To avoid a too restricted view on smaller parts of a project or a product, we selected interviewees from different parts of the development. There might however still be a selection bias as the interviewees were selected through a convenience sample through our company contacts.

1.3.3.3 External validity

These threats relate to the ability to generalize the results beyond our case studies. By design, the external validity of our studies is low. Hence, generalization of our findings to different domains or companies might not be possible. Easterbrook [46] notes that qualitative studies aim to understand and explain a given phenomenon rather than generalizing. Understanding the investigated phenomenon in one setting may help to understand other situations [36]. For

instance in Paper A, we designed our study to identify common challenges across participating companies. Thus, our research method does not support any deep reasoning about differences between companies, domains, and market positions. However, given that we found similar themes in all cases, we expect that these apply similarly to other companies or projects in large-scale systems engineering.

1.3.3.4 Reliability

Threats to reliability refer to level to which the study results are dependent on specific researchers. We limit reliability threats by improving the interview instruments in multiple iterations and by conducting interviews in pairs of two researchers. We have a prolonged involvement with our case companies and therefore a mutual trust among the parties exists. The data analysis was discussed and refined among the authors in several iterations. The results were discussed with the participating companies and also compared results obtained to available literature.

1.4 Research Synthesis

In this section, the main results and contributions of this thesis are summarized and reported per paper and main research question addressed. More detailed descriptions or the results for each study can be found in the respective paper.

1.4.1 Paper A: RE Challenges in Large-Scale Agile

RQ1: What are the RE challenges of using agile methods in large-scale systems development?
This question we answer with three sub-questions from the paper

RQ1.a.: What are possible scopes of applying agile methods in large-scale system development?

RQ1.b.: How is the role of requirements characterized in large-scale agile system development?

RQ1.c.: Which requirements related challenges exist in large-scale agile system development?

Main findings: The study revealed that large-scale companies are struggling to perform RE in agile development to the level they are used to while they were using waterfall/traditional methods. For this study, we also discovered that agile development has been adopted at different levels in different companies which also brought in a slight difference in the challenges they are facing. However, irrespective of the level of adoption, all companies exhibited challenges that were put under two particular groups: Shared understanding of User Value and Building and Maintaining System understanding. In regards to user value, most challenges were coming from teams struggling to understand customer value, writing meaningful user stories and feedback and requirements clarification from the user stories or features they develop. For system understanding, the

challenges faced relate to informing and synchronizing between teams, creating and maintaining traces, insufficient tests and user stories, agile tool chain establishment and, coming more from the traditional foundation, there was a gap between plan-driven development and agile development. Present literature concerning RE in agile development does not provide approaches for both user and system requirements specification.

Potential application of the results: Paper A contributes a map of RE related challenges in scaled agile system development. Practitioners find this map useful to plan their adoption of agile methods as well as check their process improvement, since it allows to avoid over-optimizing one aspect while negatively affecting another aspect. We hope that researchers benefit from our overview of challenges when conducting related studies. To follow up on this study, we conducted an independent study on user value (Paper C) and one addressing how agile is currently integrated in the existing processes (Paper B).

1.4.2 Paper B: Agile Integration with Existing Processes

RQ2: What challenges do agile teams face while developing in a plan-driven environment?
The answer to this question was obtained through answering the following sub-questions in Paper B

RQ2.a.: What are the perceived challenges when combining plan-driven and agile paradigms in large-scale systems engineering?

RQ2.b.: What mitigation strategies exist when using agile development in a traditional setting?

Main findings: Motivated by the findings of Paper A; challenge of gaps between agile development and plan-driven/waterfall and having 'agile islands in a waterfall', we explored this challenging aspect a bit more as understanding it would help inform the new concepts needed for agile development at scale. The findings of this paper indicate a variation in challenges faced for departments of the same company basing on the different ways of working with agile development methods. The results present challenges relating to, e.g., development teams not being aware of the high-level requirement, dealing with flexibility of writing user stories and working with later requirement changes. We found that strategies for overcoming most of these challenges are still lacking and thus call for more research.

Potential application of results: The results suggest a need for a holistic company wide approach to agile development so that some of the challenges are overcome. Although there exist studies that mention the possibility of different ways of working for sections of the same company, this study provides the proof of what challenges could ensue. From this study we also observe that while might not be possible to have all parts of the company agile nor be

desirable to have all parts of the company plan-driven, if different approaches must co-exist, one should find a way to integrate them. Future research may need to show how this integration can be achieved.

1.4.3 Paper C: User Value Addition

RQ3: How is value interpreted/viewed by the development teams in agile development?

RQ3.a.: What is the interpretation of value from the perspective of different roles in large-scale agile software development?

RQ3.b.: What are the effects of the notion of adding value every sprint of individual teams? What benefits and challenges exist?

RQ3.c.: How do you check if value has been added in each sprint?

RQ3.d.: What improvements could support adding value every sprint in large-scale continuous software engineering?

Main findings: Motivated by the findings in Paper A; challenges with shared understanding of customer value, we conducted a study to find out how development teams interpret and understand customer value on top of writing and implementing the user stories. Results show that the customer value concept in large-scale agile systems development is affected by the distance between customer and developer and the need to break down value from the whole system into manageable parts. The notion of value is fundamental for agile methods, especially for practices such as continuous delivery to the customer. From the perspective of our interviewees for this study, customer value relates to a change in the product for some while others believe it is also related to the ability to allow customers to participate and influence the discussions around new features. The change in product should relate to something a customer can sell or that makes their product or service cheaper. Customer value also relates to the relationship to the customer and become visible in development sprints as *promised features, functionality, quality, configuration, or documentation.*

Potential application of results: The results benefit practitioners in that it becomes clear how the teams are struggling with value during development, a lesson that other practitioners in similar settings can benefit from. The study also forms a basis for further evaluation of value in agile software development.

1.4.4 Paper D: Safety-Critical Systems in Agile

RQ4: What are the challenges of developing safety-critical systems (SCS) in an agile environment?
We divided this into four sub-questions in Paper D

RQ4.a.: What research exists about agile development of Safety-Critical Systems?

RQ4.b.: What are the key benefits of applying agile methods and practices in SCS development?

RQ4.c.: What challenges exist with agile development of SCS?

RQ4.d.: What solution candidates (e.g. principles and practices) promise to address challenges with respect to agile development of SCS?

Main findings: Still motivated by findings of Paper A; challenge of developing safety-critical systems in agile development, and the not so obvious stand on the current research on the topic, we venture to explore the domain through a mapping study in order to draw a map on the domain. The results obtained showed that there are reported benefits that much resemble the ones reported in the non-safety development domains. The challenges faced in safety development using agile methods are more geared towards the certification and assurance requirements of safety systems. These requirements call for well structured processes and heavy documentation that is not clearly supported by agile development and agile teams find it hard and cumbersome to balance speed and flexibility with the need for documentation.

Potential application of results: The results provide for potential generalization to other areas. We know the research that exists and that allows us and the companies to systematically search for a setup on being large-scale agile using a map of these challenges. Solutions that relate to the domain could benefit the respective industry. Results also provide a starting point for enabling us to accumulate more knowledge on which solutions can help. So future research can build on it rather than replicate it.

1.5 Conclusions and Future Work

The thesis addresses the challenges that development teams face in systems development while using agile methods, from an RE perspective. We find problems related to system understanding and getting a unified understanding of user value. Since the two concepts are interesting for development, and customer happiness is the ultimate goal of today's systems, we explored cross-cutting concerns in an attempt to address system understanding while taking customer value into consideration and thus explored the safety-critical systems development domain.

We present a map of challenges in using agile development at scale which gives us the opportunity to pull out the most pressing challenges that have not received much attention from the research industry. Thus, we explored the context of: 1) value in systems development and found challenge of addressing value coming mainly from the customer-developer distance. 2) agile adoption in systems development relating to the co-existence of both agile and plan-driven methods. Many challenges relating to tools used in order to manage the system knowledge came up. and 3) safety-critical systems development using agile methods and presented a map of challenges and what solutions are proposed in literature.

Basing on those studies, we find challenges relating to integration of agile and plan-driven methods in systems development, addressing value in large-scale development and dealing with safety in agile development. Safety is

cross-cutting between those identified challenges since the standards today still request for documentation to meet certification demands thus requiring a reliable infrastructure in which the teams can work to provide a certifiable product that is also timely and valuable to the customer.

The findings in this thesis thus conclude with a need for establishment of a beneficial infrastructure that can help inform current development and long-term support for the product. This is one such aspect to address for future research.

Bibliography

[1] U. Eliasson, R. Heldal, E. Knauss, and P. Pelliccione, "The need of complementing plan-driven requirements engineering with emerging communication: Experiences from volvo car group," in *RE Conf.* IEEE, 2015, pp. 372–381.

[2] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[3] T. Clancy, "The standish group chaos report," *Project Smart*, 2014.

[4] I. Sommerville and P. Sawyer, *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.

[5] B. Meyer, *Agile! The Good, the Hype and the Ugly*. Springer, 2014.

[6] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9, pp. 833–859, 2008.

[7] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, 2016.

[8] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Sthl, "The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson," in *ACM / IEEE Int. Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 348–356.

[9] C. Berger and U. Eklund, "Expectations and challenges from scaling agile in mechatronics-driven companies – a comparative case study," in *Proc. of 16th Int. Conf. on Agile Processes in Software Engineering and Extreme Programming (XP '15)*, 2015, pp. 15–26.

[10] M. Paasivaara and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A research proposal and a pilot study," in *Proc. of the Scientific WS Proc. of XP2016*. ACM, 2016, p. 9.

[11] V. Vinekar, C. W. Slinkman, and S. Nerur, "Can agile and traditional systems development approaches coexist? an ambidextrous view," *Information systems management*, vol. 23, no. 3, pp. 31–42, 2006.

[12] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen, "Agile software development in large organizations," *Computer*, vol. 37, no. 12, pp. 26–34, 2004.

[13] M. Laanti, O. Salo, and P. Abrahamsson, "Agile methods rapidly replacing traditional methods at nokia: A survey of opinions on agile transformation," *Information and Softw. Techn.*, vol. 53, no. 3, pp. 276–290, 2011.

[14] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development." in *WETICE*, vol. 3, 2003, p. 308.

[15] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010.

[16] B. Boehm, "Some future trends and implications for systems and software engineering processes," *Systems Engineering*, vol. 9, no. 1, pp. 1–19, 2006.

[17] J. Mössinger, "Software in automotive systems," *IEEE software*, vol. 27, no. 2, 2010.

[18] J. Dick, E. Hull, and K. Jackson, *Requirements engineering.* Springer, 2017.

[19] V. T. Heikkila, D. Damian, C. Lassenius, and M. Paasivaara, "A mapping study on requirements engineering in agile software development," in *41st Euromicro Conf. on Softw. Eng. and Advanced Applications (SEAA '15)*, 2015, pp. 199–207.

[20] V. T. Heikkilä, M. Paasivaara, C. Lasssenius, D. Damian, and C. Engblom, "Managing the requirements flow from strategy to release in large-scale agile development: a case study at ericsson," *Empirical Software Engineering*, pp. 1–45, 2017.

[21] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in human behavior*, vol. 51, pp. 915–929, 2015.

[22] A. Alliance, "Home: http://www. agilealliance. org," Accessed 14th November 2018.

[23] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for agile software development," 2001.

[24] L. Williams, "What agile teams think of agile principles," *Communications of the ACM*, vol. 55, no. 4, pp. 71–76, 2012.

[25] K. Schwaber and M. Beedle, *Agile software development with Scrum.* Prentice Hall Upper Saddle River, 2002, vol. 1.

[26] K. Beck, *Extreme programming explained: embrace change.* Addison-Wesley, 1999.

[27] E. Bjarnason, K. Wnuk, and B. Regnell, "A case study on benefits and side-effects of agile practices in large-scale requirements engineering," in *Proc. of 1st WS on Agile Reqts. Eng.*, 2011.

[28] B. Boehm, "Value-based software engineering," *SIGSOFT Softw. Eng. Notes*, vol. 28, no. 2, pp. 4–, Mar. 2003. [Online]. Available: http://doi.acm.org/10.1145/638750.638776

[29] H. H. Olsson, J. Bosch, and H. Alahyari, "Customer-specific teams for agile evolution of large-scale embedded systems," in *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2013, pp. 82–89.

[30] M. Komssi, M. Kauppinen, H. Töhönen, L. Lehtola, and A. M. Davis, "Roadmapping problems in practice: value creation from the perspective of the customers," *Requirements Engineering*, vol. 20, no. 1, pp. 45–69, 2015.

[31] S. Barney, A. Aurum, and C. Wohlin, "A product management challenge: Creating software product value through requirements selection," *Journal of Systems Architecture*, vol. 54, no. 6, pp. 576–593, 2008.

[32] C. Wohlin and A. Aurum, "Criteria for selecting software requirements to create product value: An industrial empirical study," in *Value-based software engineering*. Springer, 2006, pp. 179–200.

[33] A. Aurum and C. Wohlin, "A value-based approach in requirements engineering: explaining some of the fundamental concepts," in *Int. Working Conf. on Reqts. Eng.: Foundation for Softw. Qual. (REFSQ)*. Springer, 2007, pp. 109–115.

[34] K. Kuusinen, "Value creation and delivery in agile software development: Overcoming stakeholder conflicts," in *IFIP Conference on Human-Computer Interaction*. Springer, 2017, pp. 123–129.

[35] R. B. Woodruff, "Customer value: the next source for competitive advantage," *Journal of the academy of marketing science*, vol. 25, no. 2, pp. 139–153, 1997.

[36] H. Alahyari, R. B. Svensson, and T. Gorschek, "A study of value in agile software development organizations," *Journal of Systems and Software*, 2016.

[37] K. Schneider, *Experience and knowledge management in software engineering*. Springer Science & Business Media, 2009.

[38] T. Chau, F. Maurer, and G. Melnik, "Knowledge sharing: Agile methods vs. tayloristic methods," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*. IEEE, 2003, pp. 302–307.

[39] K. Petersen and C. Wohlin, "The effect of moving from a plan-driven to an incremental software development approach with agile practices," *Empirical Software Engineering*, vol. 15, no. 6, pp. 654–693, 2010.

[40] M. Hummel, C. Rosenkranz, and R. Holten, "The role of communication in agile systems development," *Business & Information Systems Engineering*, vol. 5, no. 5, pp. 343–355, 2013.

[41] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineering*, vol. 13, no. 3, pp. 303–337, 2008.

[42] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects–a systematic literature review," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2017, pp. 219–234.

[43] B. Fitzgerald, K.-J. Stol, R. O'Sullivan, and D. O'Brien, "Scaling agile methods to regulated environments: An industry case study," in *Proc. of 35th Int. Conf. on Software Engineering (ICSE)*, 2013, pp. 863–872.

[44] G. K. Hanssen, B. Haugset, T. Stålhane, T. Myklebust, and I. Kulbrandstad, "Quality assurance in scrum applied to safety critical software," in *Int. Conf. on Agile Software Dev.* Springer, 2016, pp. 92–103.

[45] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[46] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 285–311.

[47] P. Runeson, M. Höst, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering*, 1st ed. Wiley, 2012.

[48] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *APM&MHSR*, vol. 42, no. 5, pp. 533–544, 2015.

[49] G. R. Gibbs, *Analysing qualitative data*. Sage, 2008.

[50] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 1051–1052.

[51] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering." in *EASE*, vol. 8, 2008, pp. 68–77.

[52] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[53] T. Kahkonen, "Agile methods for large organizations-building communities of practice," in *Agile Dev. Conf., 2004*, 2004, pp. 2–10.

[54] K. Beck, *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

[55] O. Salo and P. Abrahamsson, "Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum," *IET software*, vol. 2, no. 1, pp. 58–64, 2008.

[56] U. Eklund, H. Holmström Olsson, and N. J. Strøm, "Industrial challenges of scaling agile in mass-produced embedded systems," in *Proc. of Int. WS on Agile Methods. Large-Scale Dev., Refactoring, Testing, and Estimation*, 2014, pp. 30–42.

[57] J. Pernstål, A. Magazinius, and T. Gorschek, "A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 22, no. 07, pp. 965–1004, 2012.

[58] J. Savolainen, J. Kuusela, and A. Vilavaara, "Transition to agile development-rediscovery of important requirements engineering practices," in *18th Int. Req. Eng. Conf.* IEEE, 2010, pp. 289–294.

[59] K. Wiklund, D. Sundmark, S. Eldh, and K. Lundqvist, "Impediments in agile software development: An empirical investigation," in *Proc of Product-Focused SW Process Impr.*, 2013, pp. 35–49.

[60] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, no. 6, pp. 961–971, 2008.

[61] D. Leffingwell *et al.*, "Scaled agile framework 3.0," 2014.

[62] D. Stahl and J. Bosch, "Modelling continuous integration practice differences in industry software development," *Systems and Software*, vol. 87, pp. 48–59, 2014.

[63] ISO, "Road vehicles – Functional safety," 2011.

[64] R. Hoda, J. Noble, and S. Marshall, "Self-organizing roles on agile software development teams," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 422–444, 2013.

[65] F. Evbota, E. Knauss, and A. Sandberg, "Scaling up the planning game: Collaboration challenges in large-scale agile product development," in *Proc. of 17th Int'l Conf. on Agile Softw. Dev. (XP)*, Edinburgh, UK, 2016.

[66] R. Kasauli, E. Knauss, A. Nilsson, and S. Klug, "Adding value every sprint: A case study on large-scale continuous requirements engineering," in *Proc. of 3rd WS on Cont. Reqts. Eng.*, Essen, Germany, 2017.

[67] E. Bjarnason, M. Unterkalmsteiner, M. Borg, and E. Engström, "A multi-case study of agile requirements engineering and the use of test cases as requirements," *Information and Software Technology*, vol. 77, pp. 61–79, 2016.

[68] R. Wohlrab, J.-P. Steghöfer, E. Knauss, S. Maro, and A. Anjorin, "Collaborative traceability management: Challenges and opportunities," in *Proc. of 24th Int. Reqts. Eng. Conf. (RE)*, 2016, pp. 216–225.

[69] G. Van Waardenburg and H. Van Vliet, "When agile meets the enterprise," *Information and software technology*, vol. 55, no. 12, pp. 2154–2171, 2013.

[70] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, "Requirements engineering challenges in large-scale agile system development," in *25th Int. Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 352–361.

[71] R. Wohlrab, P. Pelliccione, E. Knauss, and S. C. Gregory, "The problem of consolidating re practices at scale: An ethnographic study," in *REFSQ*. Springer, 2018, pp. 155–170.

[72] F. Almeida, "Challenges in migration from waterfall to agile environments," *World Journal of Computer Application and Technology*, vol. 5, no. 3, pp. 39–49, 2017.

[73] S. Bannink, "Challenges in the transition from waterfall to scrum–a casestudy at portbase," in *20th Twente Student Conference on Information Technology*, 2014.

[74] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, "Is water-scrumfall reality? on the use of agile and traditional development practices," in *Int. Conf. on Product-Focused Software Process Improvement*, 2015, pp. 149–166.

[75] R. J. Kusters, Y. van de Leur, W. G. Rutten, and J. J. Trienekens, "When agile meets waterfall - investigating risks and problems on the interface between agile and traditional software development in a hybrid development organization." in *Int. Conf. on Enterprise Information Systems (ICEIS)*, vol. 2, 2017, pp. 271–278.

[76] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E. Hanser *et al.*, "Hybrid software and system development in practice: waterfall, scrum, and beyond," in *ICSSP*. ACM, 2017, pp. 30–39.

[77] K. Kuusinen, P. Gregory, H. Sharp, and L. Barroca, "Strategies for doing agile in a non-agile environment," in *ESEM*. ACM, 2016, p. 5.

[78] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE software*, vol. 22, no. 5, pp. 30–39, 2005.

[79] A. Sillitti and G. Succi, "Requirements engineering for agile methods," in *Engineering and Managing Software Requirements*. Springer, 2005, pp. 309–326.

[80] R. S. Carson, "4.2. 1 can systems engineering be agile? development lifecycles for systems, hardware, and software," in *INCOSE International Symposium*, vol. 23, no. 1. Wiley Online Library, 2013, pp. 16–28.

[81] R. K. Yin, "Case study research: Design and methods 4th ed," in *United States: Library of Congress Cataloguing-in-Publication Data*, vol. 2, 2009.

[82] S. Gopakumar, "Challenges and strategies for balancing plan-driven and agile requirement engineering," 2016.

[83] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, and G. Stieglbauer, "Organisation and communication problems in automotive requirements engineering," *Requirements Engineering*, vol. 23, no. 1, pp. 145–167, 2018.

[84] E. J. Uusitalo, M. Komssi, M. Kauppinen, and A. M. Davis, "Linking requirements and testing in practice," in *16th RE Conf.* IEEE, 2008, pp. 265–270.

[85] F. G. de Oliveira Neto, J. Horkoff, E. Knauss, R. Kasauli, and G. Liebel, "Challenges of aligning requirements engineering and system testing in large-scale agile: A multiple case study," in *REW*. IEEE, 2017, pp. 315–322.

[86] D. Cohen, M. Lindvall, and P. Costa, "Agile software development," DACS SOAR Report, Tech. Rep., 2003.

[87] K. Schwaber, *Agile project management with Scrum*. Microsoft Press, 2004.

[88] M. Fowler, "Continuous integration," Tech. Rep., 2006, http://martinfowler.com/articles/continuousIntegration.html last visit: 01 2017.

[89] D. Leffingwell, *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley Professional, 2011.

[90] D. Reifer, F. Maurer, and Erdogmus, "Scaling agile methods," *IEEE Software*, vol. 20, no. 4, pp. 12–14, 2001.

[91] S. Neely and S. Stolt, "Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy)," in *Proc. of Agile Conference*. Nashville TN, USA: IEEE, 2013, pp. 121–128.

[92] T. Dingsøyr and N. B. Moe, "Towards principles of large-scale agile development," in *International Conference on Agile Software Development*. Springer, 2014, pp. 1–8.

[93] M. Roberts, "Enterprise continuous integration using binary dependencies," in *Extreme Progr. and Agile Proc. in Softw. Eng. (XP '04)*. Springer, 2004, pp. 194–201.

[94] R. Rogers, "Scaling continuous integration," in *Extreme Programming and Agile Processes in Software Engineering*. Springer, 2004, pp. 68–76.

[95] A. Debbiche, M. Diener, and R. B. Svensson, "Challenges when adopting continuous integration: A case study," in *Proc. of 15th Int. Conf. of Product Focused SW Dev. and Process Impr. (Profes)*, Helsinki, Finland, 2014, pp. 17–32.

[96] F. R. Golra, A. Beugnard, F. Dagnat, S. Guerin, and C. Guychard, "Continuous requirements engineering using model federation," in *Proc. of 24th Int. Reqts. Eng. Conf. (RE)*. IEEE, 2016, pp. 347–352.

[97] M. Kirikova, "Continuous requirements engineering in freedom framework: A position paper," in *Proc. of 2nd WS on Cont. Reqts. Eng. (CRE)*, Gothenburg, Sweden, 2016.

[98] S. Berczuk, "Back to basics: The role of agile principles in success with an distributed scrum team," in *Agile Conference (AGILE)*, 2007, pp. 382–388.

[99] M. Kauppinen, J. Savolainen, L. Lehtola, M. Komssi, H. Tohonen, and A. Davis, "From feature development to customer value creation," in *Proc. of 17th IEEE Int. Reqts. Eng. Conf. (RE)*. IEEE, 2009, pp. 275–280.

[100] Z. Racheva, M. Daneva, K. Sikkel, A. Herrmann, and R. Wieringa, "Do we know enough about requirements prioritization in agile projects: Insights from a case study," in *Proc. of 18th Int. Requts Eng. Conf. (RE 10)*, Sydney, Australia, 2010.

[101] L. Mathiassen, "Collaborative practice research," *Information Technology and People*, vol. 15, no. 4, pp. 321–345, 2002.

[102] H. S. Neap and T. Celik, "Value of a product: A definition," *Int. Journal of Value-Based Management*, vol. 12, no. 2, pp. 181–191, 1999.

[103] P. A. Laplante and J. F. DeFranco, "Software engineering of safety-critical systems: Themes from practitioners," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 825–836, 2017.

[104] J. Hatcliff, A. Wassyng, T. Kelly, C. Comar, and P. Jones, "Certifiably safe software-dependent systems: challenges and directions," in *Proceedings of the on Future of Software Engineering*. ACM, 2014, pp. 182–200.

[105] L. Provenzano and K. Hänninen, "Specifying software requirements for safety-critical railway systems: An experience report," in *Int. Working Conf. on Requirements Engineering: Foundation for Software Quality*. Springer, 2017, pp. 363–369.

[106] E. S. Grant, "Requirements engineering for safety critical systems: An approach for avionic systems," in *2nd Int. Conf. on Computer and Communications (ICCC), 2016*. IEEE, 2016, pp. 991–995.

[107] M. Vuori, "Agile development of safety-critical software," *Tampere University of Technology*, vol. 14, 2011.

[108] J. P. Notander, M. Höst, and P. Runeson, "Challenges in flexible safety-critical software development an industrial qualitative survey," in *Proc. of Int. Conf. on Product Focused Software Process Improvement (PROFES)*, 2013, pp. 283–297.

[109] X. Ge, R. F. Paige, and J. A. McDermid, "An iterative approach for development of safety-critical software and safety arguments," in *Proc. of AGILE Conf.* Nashville, TN, USA: IEEE, 2010, pp. 35–43.

[110] P. Pelliccione, E. Knauss, and et al., "Automotive architecture framework: the experience of volvo cars," *Journal of systems architecture*, 2017.

[111] M. Kaisti, V. Rantala, T. Mujunen, S. Hyrynsalmi, K. Könnölä, T. Mäkilä, and T. Lehtonen, "Agile methods for embedded systems development-a literature review and a mapping study," *EURASIP Journal on Embedded Systems*, vol. 2013, no. 1, p. 15, 2013.

[112] O. Cawley, X. Wang, and I. Richardson, "Lean/agile software development methodologies in regulated environments–state of the art," in *Proc. of Conf. on Lean Enterprise Software and Systems (LESS)*, Helsinki, Finland, 2010, pp. 31–36.

[113] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," in *Technical report, Ver. 2.3. EBSE*. sn, 2007.

[114] P. Diebold and M. Dahlem, "Agile practices in practice: A mapping study," in *Proc. of the 18th Int. Conf. on Evaluation and Assessment in Software Engineering*, ser. EASE '14, 2014, pp. 30:1–30:10.

[115] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 2014, p. 38.

[116] K. Łukasiewicz and J. Górski, "Agilesafe - a method of introducing agile practices into safety-critical software development processes," in *Proc. of Federated Conf. on Computer Science and Information Systems (FedCSIS)*, 2016, pp. 1549–1552.

[117] T. Stålhane and T. Myklebust, "Early safety analysis," in *Proceedings of the Scientific Workshop Proceedings of XP2016*. ACM, 2016, p. 19.

[118] J. Schmidt and K. Weyrauch, "Getting agile with medical device development," *Biomedical Instrumentation & Technology*, vol. 47, no. 3, pp. 221–223, 2013.

[119] Y. Wang and S. Wagner, "Toward integrating a system theoretic safety analysis in an agile development process." in *Proc. of Software Engineering (Workshops)*, Wien, Austria, 2016, pp. 156–159.

[120] T. Myklebust, T. Stålhane, and N. Lyngby, "An agile development process for petrochemical safety conformant software," in *Proc. of Annual Reliability and Maintainability Symposium (RAMS)*, Tucson, AZ, USA, 2016.

[121] G. K. Hanssen, G. Wedzinga, and M. Stuip, "An assessment of avionics software development practice: Justifications for an agile development process," in *Proc. of Int. Conf. on Agile Software Dev. (XP)*, Cologne, Germany, 2017, pp. 217–231.

[122] L. T. Heeager, "How can agile and documentation-driven methods be meshed in practice?" in *(XP)*, Rome, Italy, 2014, pp. 62–77.

[123] R. Rasmussen, T. Hughes, J. Jenks, and J. Skach, "Adopting agile in an fda regulated environment," in *Proc. of AGILE Conf.*, Chicago, IL, USA, 2009, pp. 151–155.

[124] T. Stålhane, G. K. Hanssen, T. Myklebust, and B. Haugset, "Agile change impact analysis of safety critical software," in *Proc. of Int. Conf. on Computer Safety, Reliability, and Security (SAFECOMP)*, Firenze, Italy, 2014, pp. 444–454.

[125] S. Vost and S. Wagner, "Keeping continuous deliveries safe," in *Proc. of 39th Int. Conf. on SW Eng. (ICSE)*, Buenos Aires, Argentina, 2017.

[126] M. McHugh, F. McCaffery, and G. Coady, "An agile implementation within a medical device software organisation," in *(SPICE)*, 2014, pp. 190–201.

[127] J. Górski and K. Łukasiewicz, "Assessment of risks introduced to safety critical software by agile practices-a software engineer's perspective," *Computer Science (CSCI)*, vol. 13, no. 4, pp. 165–182, 2012.

[128] P. A. Rottier and V. Rodrigues, "Agile development in a medical device company," in *Proc. of AGILE Conf.*, Toronto, ON, Canada, 2008, pp. 218–223.

[129] S. Wolff, "Scrum goes formal: Agile methods for safety-critical systems," in *Formal Methods in Software Engineering: Rigorous and Agile Approaches (FormSERA)*, Zurich, Switzerland, 2012, pp. 23–29.

[130] R. F. Paige, A. Galloway, R. Charalambous, X. Ge, and P. J. Brooke, "High-integrity agile processes for the development of safety critical software," *Int. Journal of Critical Computer-Based Systems (IJCCBS)*, vol. 2, no. 2, pp. 181–216, 2011.

[131] Y. Wang, J. Ramadani, and S. Wagner, "An exploratory study on applying a scrum development process for safety-critical systems," in *Proc. of Int. Conf. on Product-Focused Software Process Improvement (PROFES)*, 2017, pp. 324–340.

[132] Y. Wang, I. Bogicevic, and S. Wagner, "A study of safety documentation in a scrum development process," in *Proc. of XP Scientific Workshops (XP WS)*, 2017.

[133] H. Jonsson, S. Larsson, and S. Punnekkat, "Agile practices in regulated railway software development," in *Proc. of 23rd Int. Symp. on Software Reliability Engineering, Workshops (ISSREW WS)*, Dallas, TX, USA, 2012, pp. 355–360.

[134] W. Kuchinke, C. Krauth, and T. Karakoyun, "Agile software development requires an agile approach for computer system validation of clinical trials software products," in *Proc. of eChallenges Conf.*, Belfast, UK, 2014, pp. 1–8.

[135] M. McHugh, F. McCaffery, and V. Casey, "Barriers to adopting agile practices when developing medical device software," in *Proc. of Int. Conf. on SW Process Impr. and Capability Determ. (SPICE)*, Palma de Mallorca, Spain, 2012, pp. 141–147.

[136] K. Trektere, F. McCaffery, M. Lepmets, and G. Barry, "Tailoring mdevspice for mobile medical apps," in *Software and System Processes (ICSSP)*, Austin (TX), USA, 2016, pp. 106–110.

[137] O. Doss and T. Kelly, "Addressing the 4+1 software assurance processes within scrum," in *Proc. of XP Scientific Workshops*, Edinburgh, Scotland, UK, 2016, 5 pages.

[138] T. Stålhane and T. Myklebust, "The agile safety case," in *Proc. of Int. Conf. on Computer Safety, Reliability, and Security (SAFECOMP)*, 2016, pp. 5–16.

[139] A. Wils, S. V. Baelen, T. Holvoet, and K. D. Vlaminck, "Agility in the avionics software world," in *Proc. of (XP)*, 2006, pp. 123–132.

[140] O. Doss, T. Kelly, T. Stålhane, B. Haugset, and M. Dixon, "Integration of the 4+1 software safety assurance principles with scrum," in *Proc. of European Conf. on Software Process Improvement (EuroSPI)*, Ostrava, Czech Republic, 2017, pp. 72–82.

[141] A. Abdelaziz, Y. El-Tahir, and R. Osman, "Adaptive software development for developing safety critical software," in *Proc. of Int. Conf. on Computing, Control, Networking, Electronics and Embedded Systems Eng. (ICCNEEE)*, Khartoum, Sudan, 2015, pp. 41–46.

[142] J. Górski and K. Łukasiewicz, "Towards agile development of critical software," in *In Proc. of Int. Workshop on Software Engineering for Resilient Systems (SERENE)*, 2013, pp. 48–55.

[143] P. E. McMahon, "Cmmi the agile way in constrained and regulated environments," *Journal of Defense Software Engineering (Crosstalk)*, vol. JulAug, pp. 10–15, 2016.

[144] K. Gary, A. Enquobahrie, L. Ibanez, P. Cheng, Z. Yaniv, K. Cleary, S. Kokoori, B. Muffih, and J. Heidenreich, "Agile methods for open source safety-critical software," *Journal of Software: Practice and Experience (SPE)*, vol. 41, no. 9, pp. 945–962, 2011.